UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/729,508	12/04/2000	Rene Vangemert	00-487/1496.00053	9954

24319 7590 01/13/2005

LSI LOGIC CORPORATION
1621 BARBER LANE
MS: D-106
MILPITAS, CA 95035

EXAMINER	
GERSTL, SHANE F	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 01/13/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/729,508	VANGEMERT ET AL.	
	Examiner	Art Unit	
	Shane F Gerstl	2183	

-- The MAILING DATE of this communication appears n th cover sheet with the corresp ndence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 October 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *4,5,11,12,17,18 and 23* is/are allowed.

6)☒ Claim(s) *1-3,6-10,13-16 and 19-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-23 have been examined.

Papers Received

2. Receipt is acknowledged of amendment papers submitted, where the papers have been placed of record in the file.

3. The amendment filed 28 October 2004 has successfully overcome the objections to the claims whereby the objections have been withdrawn.

Claim Rejections - 35 USC § 112

4. Claims 3 and 10 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

5. Claims 3 and 10 seem to present an impossible scenario. The claims read that the first status is set to the invalid state in response to the first register receiving an invalid second operand data prior to transferring the second operand data to the first register. It cannot be understood how a status can be set in response to having received data, yet be set before the data is transferred. Thus the Examiner does not know how to interpret the claim.

Claim Rejections - 35 USC § 102

6. Claims 1, 15, 16, and 21 are rejected under 35 U.S.C. 102(b) as being anticipated by Mirapuri (5,590,294).

7. In regard to claim 1, Mirapuri discloses a method of recovering from invalid data in a first register within a pipelined processor (figure 5 shows pipelining), the method comprising the steps of:

a. setting a first status unique to a first operand data stored in said first register to an invalid state in response to said first operand data being invalid; [In column 9, lines 35-42, Mirapuri discloses a set of control registers that tell whether a pipeline stage contains a valid or invalid instruction. Because the pipelined processor given by Mirapuri has eight stages and takes eight clock cycles to complete one instruction (column 5, lines 16-18), there are therefore at least eight pipeline registers. These registers hold the data for each stage that the control registers give the status of. Column 10, lines 56-58, show that data is loaded as invalid: "Often, the stall condition is only detected after parts of the pipeline have advanced using incorrect data." The included dictionary definition of "unique" (definition 1) shows that to be unique is to be the only one or the sole. Thus the claim language that recites "a first status unique to a first operand data stored in said first register" can be read as a first status of a first operand data stored in said first register, which is the sole status of this type for this operand. With this interpretation one needs only show that Mirapuri discloses that each pipeline stage has only a single status for the instruction (including the operands) therein. As shown in column 9, lines 20-65 Mirapuri only discloses a single chain of registers, one for each stage, indicating the validity of the stage and the

instructions and operands therein. Thus, this status which is unique to the operand in what it conveys is disclosed by Mirapuri.]

b. stalling said processor in response to both (i) an instruction requiring said first operand data and (ii) said first status being in said invalid state. [As shown above from column 10, lines 56-58, a stall condition is detected after invalid data has advanced, showing that the data must be needed since validity was not detected until now. Then after detecting the stall condition, Mirapuri stalls.]

8. In regard to claim 15, Mirapuri discloses a pipelined processor (figure 5 shows pipelining) comprising:

a. means for buffering a first operand data; [The pipelined processor given by Mirapuri has eight stages and takes eight clock cycles to complete one instruction (column 5, lines 16-18), there are therefore at least eight pipeline registers. These registers buffer data to the next stage of the pipeline.]

b. means for buffering a first status uique to said first operand data; [In column 9, lines 35-42, Mirapuri discloses a set of control registers that that tell whether a pipeline stage contains a valid or invalid instruction. The pipeline registers mentioned above contain the valid or invalid instruction. Thus, the control registers buffer the status of the pipeline registers. The included dictionary definition of "unique" (definition 1) shows that to be unique is to be the only one or the sole. Thus the claim language that recites "a first status unique to a first operand data stored in said first register" can be read as a first status of a first operand data stored in said first register, which is the sole status of this

type for this operand. With this interpretation one needs only show that Mirapuri discloses that each pipeline stage has only a single status for the instruction (including the operands) therein. As shown in column 9, lines 20-65 Mirapuri only discloses a single chain of registers, one for each stage, indicating the validity of the stage and the instructions and operands therein. Thus, this status which is unique to the operand in what it conveys is disclosed by Mirapuri.]

c. means for setting said first status to an invalid state in response to said first operand data being invalid; [It is inherent that if the status registers of mention above hold validity data, then there is a means for setting the status. Column 10, lines 56-58, show that data is loaded as invalid when it says, "Often, the stall condition is only detected after parts of the pipeline have advanced using incorrect data."]

d. means for stalling said processor in response to both (i) an instruction requiring said first operand data and (ii) said first status being in said invalid state. [In column 6, lines 43-55, Mirapuri discloses the case of an instruction cache miss. Mirapuri states here that if there is a miss, an interlock occurs resulting in a stall. This miss detection is shown to not occur until after the fetch and decode stages (lines 49-52). Therefore, the pipeline registers of the fetch and decode stage are invalid when a miss occurs as Mirapuri shows, "Such stages are invalid at the time of the interlock," (column 10, lines 63-64).]

9. In regard to claim 16, Mirapuri discloses the method of claim 1, further comprising the step of: setting said first register status to said invalid state in response

to a miss for a read from a data cache of a second data to be written in said first register. [Column 5, lines 29-35 show that data is always fetched from a cache in the pipeline since the cache has dedicated stages for fetching from it. Therefore, if the first register is in one of the cache stages, data of the instruction is fetched from the data cache for storage in the pipeline registers so it can be processed. Column 6, lines 8-11 show that cache misses cause interlocks. As shown previously, an interlock is caused by an invalid state of a pipeline stage and thus the cache miss sets the control register to invalid.]

10. In regard to claim 21, Mirapuri discloses the method according to claim 1, further comprising the step of: continuing operation in a stage of said pipelined processor containing said first operand data in response to no instruction requiring said first operand data while said first status is in said invalid state. [The Examiner is interpreting the phrase "no instruction requiring said first operand data while said first status is in said invalid state" to mean that no instruction needs the first operand data to be in the invalid state in order to execute. With this interpretation, it is inherent that the operation in a stage containing the first operand will continue when the instruction therein needs only valid data since there would be no reason to halt operation.]

Claim Rejections - 35 USC § 103

11. Claims 2 and 7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mirapuri in view of Sites (5,193,167).

12. In regard to claim 2,

a. Mirapuri discloses the method of claim 1, further comprising the step of: setting said first status to said invalid state (as described above).

b. Mirapuri does not disclose the above function in response to a conditional store for said first operand data in said first register.

c. Sites has taught in column 3, lines 47-56 the use of a conditional store instruction. This instruction is shown to correspond to a matching locked load instruction. The store updates a memory position from the register loaded by the locked load instruction only if the register was not written to since the locked load instruction. In other words, the store conditional stores the contents of a register into memory based on the register's validity. This means that if the register was written to, the data in the register is not written to memory. Therefore, invalid data was loaded into the register. When one incorporates this disclosure of Sites into the design of Mirapuri, one of ordinary skill in the art can see that when a conditional store is encountered and it is discovered that invalid data was loaded into the register, the status of the register would be set to invalid as shown above.

d. Sites has taught that this conditional store instruction permits the use of atomic byte writes (lines 47-49). This means that outside processes can only access the data in use before and after the transaction is complete. Otherwise, the store is not carried out. This data integrity that is provided during the execution of the conditional store instruction would have motivated one of

ordinary skill in the art to implement the conditional store instruction taught by Sites in the design of Mirapuri.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri to include the conditional store instruction taught by Sites in order to achieve data integrity upon executing a store.

13. In regard to claim 7, Mirapuri has disclosed the method of claim 1, as described above, further comprising the steps of:

a. setting said first status to said invalid state in response to at least one of (i) a conditional store for said first data in said first register, (ii) receiving a second data from a second register having a second status in said invalid state and (iii) a miss for a data cache read of a third data to be written in said first register.

i. Mirapuri discloses the method of claim 1, further comprising the step of: setting said register status to said invalid state (as described above).

ii. Mirapuri does not disclose that said register status for said register is set to said invalid state in response to a conditional store for said register.

iii. Sites has taught in column 3, lines 47-56 the use of a conditional store instruction. This instruction is shown to correspond to a matching locked load instruction. The store updates a memory position from the register loaded by the locked load instruction only if the register was not written to since the locked load instruction. In other words, the store

conditional stores the contents of a register into memory based on the register's validity. This means that if the register was written to, the data in the register is not written to memory. Therefore, invalid data was loaded into the register. When one incorporates this disclosure of Sites into the design of Mirapuri, one of ordinary skill in the art can see that when a conditional store is encountered and it is discovered that invalid data was loaded into the register, the status of the register would be set to invalid as shown above.

iv. Sites has taught that this instruction pair permits the use of atomic byte writes (lines 47-49). This means that outside processes can only access the data in use before and after the transaction is complete. Otherwise, the store is not carried out. This data integrity that is provided during the execution of the conditional store instruction would have motivated one of ordinary skill in the art to implement the conditional store instruction taught by Sites in the design of Mirapuri.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri to include the conditional store instruction taught by Sites in order to achieve data integrity upon executing a store.

b. obtaining said third operand data from a memory in response to at least one of said conditional store and said miss; [Since the previous section makes

use of three limitations in the alternative and the first limitation has been met, there is no third data and this limitation is moot.]

c. stalling said processor in response to obtaining said third operand data to enable said third operand data to be written in said first register; [Since a previous section makes use of three limitations in the alternative and the first limitation has been met, there is no third data and this limitation is moot.]

14. Claims 8, 13, 19-20, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mirapuri in view of Steiss (5,815,420).

15. In regard to claim 8,

a. Mirapuri discloses a pipelined processor (figure 5 shows pipelining), comprising:

i. A first register configured to buffer a first operand data

ii. logic configured to

set said first status for said register to an invalid state in response to said first operand data being invalid; [In column 9, lines 35-42, Mirapuri discloses a set of control registers that that tell whether a pipeline stage contains a valid or invalid instruction. Because the pipelined processor given by Mirapuri has eight stages and takes eight clock cycles to complete one instruction (column 5, lines 16-18), there are therefore at least eight pipeline registers. These registers hold the data for each stage that the control registers give the status of. Column 10, lines 56-58, show that data

is loaded as invalid: "Often, the stall condition is only detected after parts of the pipeline have advanced using incorrect data."]

stall said processor in response to both (a) an instruction requiring said first data and (b) said first status being in said invalid state. [As shown above from column 10, lines 56-58, a stall condition is detected after invalid data has advanced, showing that the data must be needed since validity was not detected until now. Then after detecting the stall condition, Mirapuri stalls.]

b. Mirapuri does not disclose that the register is configured to buffer a first status unique to said first operand data. Mirapuri keeps track of register status in control registers that contain pipeline register validity information. The included dictionary definition of "unique" (definition 1) shows that to be unique is to be the only one or the sole. Thus the claim language that recites "a first status unique to a first operand data stored in said first register" can be read as a first status of a first operand data stored in said first register, which is the sole status of this type for this operand. With this interpretation one needs only show that Mirapuri discloses that each pipeline stage has only a single status for the instruction (including the operands) therein. As shown in column 9, lines 20-65 Mirapuri only discloses a single chain of registers, one for each stage, indicating the validity of the stage and the instructions and operands therein. Thus, this status which is unique to the operand in what it conveys is disclosed by Mirapuri.

c. Steiss has disclosed a register configured to buffer a register status (figure 2, V bit). This status is used in the same manner as disclosed by Mirapuri for stalling the processor (Steiss, column 10, lines 9-11).

d. Having the register status bits buffered by the registers themselves allows for closer physical access of validity information to its corresponding data. One of ordinary skill in the art would have recognized the advantages of a central location for all of this data and thus would have been motivated to move the validity status bits into the pipeline register. This physical convenience and speedup would have motivated one of ordinary skill in the art at the time of invention to integrate the validity information of the registers in Mirapuri's disclosure into the register itself as described by Steiss. It is completely proper that one of ordinary skill in the art would expect some sort of speedup from data all being in a central location. This is because the close proximity of the data to the logic that uses it (rather than having the status bits further away from the logic) results in shorter wire lengths for connection. Sending something over a shorter distance is inherently faster than sending something over a longer distance. In addition, one of ordinary skill in the art would have recognized that this close proximity improves area on the chip die since the elements are more compact. Further, the courts have found that shifting the location of parts without modifying the function of a device is not a patentable distinction. Thus in this case simply moving the status bits into the register with the data is not a patentable distinction. See In re Japikse 86 USPQ 70 (CCPA1950).

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri to include the register status information in the register itself as given by Steiss for the purposes of convenience in location, logical speedup, and improved die area.

16. In regard to claim 13,

a. Mirapuri in view of Steiss, as applied to claim 8, discloses the pipelined processor of claim 8, as described above, that includes the register status in control registers that are checked on each clock cycle for indicating invalid states of pipeline stages;

b. Mirapuri in view of Steiss, as applied to claim 8, does not disclose that the register status is buffered as a plurality of bits to provide for multiple conditions that would set said register status to said invalid state.

c. Steiss has disclosed a pipelined processor (figure 1) that keeps track of register status. Steiss has disclosed that register status is buffered as a plurality of bits (figure 3, element 51) to provide for multiple conditions that would set said register status to said invalid state. In column 6, lines 50-56, Steiss shows the operation of the mentioned buffer as holding the status of multiple operands (conditions within the pipeline stage). In column 10, lines 9-11, Steiss has shown that these valid bits are used for stalling the processor, as is so in the disclosure of Mirapuri in view of Steiss as applied to claim 8.

d. Implementing the multiple bit validity buffer of Steiss in the design of Mirapuri in view of Steiss, as applied to claim 8, would allow each pipeline stage

to realize invalid information for multiple conditions based on different operands. Upon realizing that the stage has received invalid data, the pipeline register would signify invalid data as described above. The ability to detect multiple reasons for invalidity would have motivated one of ordinary skill in the art to integrate the design of Steiss' validity buffer into that of Mirapuri in view of Steiss, as applied to claim 8.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri in view of Steiss, as applied to claim 8, to include the validity buffer of Steiss in order to obtain the ability to detect multiple reasons for pipeline stage invalidity.

17. In regard to claim 19, Mirapuri in view of Steiss discloses the pipelined processor of claim 13, wherein said logic comprises a first logic gate configured to combine said bits of said first status read from said first register. [Steiss has shown in figure 3 (with column 6, lines 43-64) that the status bits are combined into a single buffer 51.]

18. In regard to claim 20, Mirapuri in view of Steiss discloses the pipeline processor of claim 8, wherein said logic comprises a first logic gate configured to combine a plurality of bits to form said first status written to said first register. [Column 6, lines 26-29 show that the valid bit is set based on a comparison of values in an entry R. It is inherent that this comparison comprises a logic gate, and perhaps multiple gates, that combine the bits of the values being compared. This comparison serves as the basis for forming the first status written to the first register.]

19. In regard to claim 22, Mirapuri in view of Steiss discloses the pipelined processor of claim 8, wherein said pipelined processor is configured to continue operating in a stage containing said first operand data in response to no instruction requiring said first operand data while said first status is in the invalid state. [The Examiner is interpreting the phrase "no instruction requiring said first operand data while said first status is in said invalid state" to mean that no instruction needs the first operand data to be in the invalid state in order to execute. With this interpretation, it is inherent that the operation in a stage containing the first operand will continue when the instruction therein needs only valid data since there would be no reason to halt operation.]

20. Claims 9 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mirapuri in view of Steiss as applied to claim 8 above, and further in view of Sites.

21. In regard to claim 9,

a. Mirapuri in view of Steiss discloses the pipelined processor of claim 8, as described above, wherein said logic being further configured to set said first status to said invalid state (as described above).

b. Mirapuri in view of Steiss does not disclose the above function in response to a conditional store for said first operand data in said first register.

c. Sites has taught in column 3, lines 47-56 the use of a conditional store instruction. This instruction is shown to correspond to a matching locked load instruction. The store updates a memory position from the register loaded by the locked load instruction only if the register was not written to since the locked load instruction. In other words, the store conditional stores the contents of a register

into memory based on the register's validity. This means that if the register was written to, the data in the register is not written to memory. Therefore, invalid data was loaded into the register. When one incorporates this disclosure of Sites into the design of Mirapuri, one of ordinary skill in the art can see that when a conditional store is encountered and it is discovered that invalid data was loaded into the register, the status of the register would be set to invalid as shown above.

d. Sites has taught that this instruction pair permits the use of atomic byte writes (lines 47-49). This means that outside processes can only access the data in use before and after the transaction is complete. Otherwise, the store is not carried out. This data integrity that is provided during the execution of the conditional store instruction would have motivated one of ordinary skill in the art to implement the conditional store instruction pair taught by Sites in the design of Mirapuri in view of Steiss.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri in view of Steiss to include the conditional store instruction pair taught by Sites in order to achieve data integrity upon executing a store.

22. In regard to claim 14, Mirapuri in view of Steiss has disclosed the pipelined processor of claim 8, as described above, further comprising:

a. A bus interface unit (figure 4, element 14) configured to obtain a third operand data for said first register from a memory; [The memory management unit controls accesses to and from the memory thus controlling the bus and

acting as a bus interface unit. As described above, invalid data is at times loaded into the registers and when a register is invalid and accessed, the processor stalls. Mirapuri has shown in column 10, lines 63-65 that stages invalid at the interlock (stall) are preloaded with correct information. Because Mirapuri has shown in column 6, lines 54-56, one of multiple cases where correct data is loaded from memory, it has been shown that Mirapuri in view of Steiss loads correct information from memory when invalid data exists.]

b. said logic is further configured to set said first status for said register to said invalid state in response to a conditional store for said first operand data in said first register.

i. Mirapuri discloses the method of claim 1, as described above, further comprising the step of: setting said register status to said invalid state (as described above).

ii. Mirapuri does not disclose the above function in response to a conditional store for said register.

iii. Sites has taught in column 3, lines 47-56 the use of a conditional store instruction. This instruction is shown to correspond to a matching locked load instruction. The store updates a memory position from the register loaded by the locked load instruction only if the register was not written to since the locked load instruction. In other words, the store conditional stores the contents of a register into memory based on the register's validity. This means that if the register was written to, the data

in the register is not written to memory. Therefore, invalid data was loaded into the register. When one incorporates this disclosure of Sites into the design of Mirapuri, one of ordinary skill in the art can see that when a conditional store is encountered and it is discovered that invalid data was loaded into the register, the status of the register would be set to invalid as shown above.

iv. Sites has taught that this instruction pair permits the use of atomic byte writes (lines 47-49). This means that outside processes can only access the data in use before and after the transaction is complete. Otherwise, the store is not carried out. This data integrity that is provided during the execution of the conditional store instruction would have motivated one of ordinary skill in the art to implement the conditional store instruction pair taught by Sites in the design of Mirapuri.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri to include the conditional store instruction pair taught by Sites in order to achieve data integrity upon executing a store.

c. A second register; set said first status for said register to said invalid state in response to receiving a second data from said second register having a second status in said invalid state. [The case given above for claim 1 shows that if the pipeline has advanced invalid data, then whatever stage where the invalid

data was detected has received its data from the previous pipeline stage and register.]

d. stall said processor in response to obtaining said third operand data for said first register; [In column 10, lines 59-63, Mirapuri has shown that parts of the pipeline are frozen (stalled) while others obtain corrected information. After receiving the corrected data, the pipeline registers of these stages are redone, but, the stalled stages continue to stall, thus in response to receiving valid data.]

e. write said third operand data in said first in response to stalling said processor. [As just described, when parts of the pipeline are stalled and others receive data, that data is then loaded to the pipeline registers.]

f. set said first status to said invalid state in response to a cache load-miss for said first register. [In column 6, lines 43-54, Mirapuri discloses the case of a cache miss. Mirapuri shows here that in such a case, the processor stalls. As shown above, the pipeline register is set to invalid when correct data is received by the bus interface unit.]

23. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mirapuri in view of Blomgren (5,542,109).

24. In regard to claim 6,

a. Mirapuri discloses the method of claim 1 as shown previously.

b. Mirapuri does not disclose that the first status is buffered as a plurality of bits to provide for multiple conditions that would indicate said invalid state.

c. Blomgren has disclosed that the first status is buffered as a plurality of bits (figure 5; 1st, 2nd, 3rd valid entries) to provide for multiple conditions that would indicate said invalid state. Column 12, lines 23-35 show that the valid bits are encoded into first second and third fields each for representing an instruction. Table 2 in column 11 shows that each of the first two fields is composed of a plurality of bits that indicate multiple conditions. Column 11, lines 28-34 show that the valid fields are for indicating valid instructions (functional control words) in each stage of the pipeline. Column 3, lines 15-21 show that the valid bits indicate locations and order of valid instructions in the pipelines as a part of an address tracking means.

d. Blomgren has shown in column 1, line 65 – column 2, line 43 that address tracking is used for exception recovery and that the disclosed invention does so without storing each address in every stage, but, instead determines them whenever needed (using the valid bits shown above) and one of ordinary skill in the art would have recognized that this is low-cost since fewer resourced are consumed yielding less power consumption. This ability to track instruction addresses at a low-cost would have motivated one of ordinary skill in the art to modify the design of Mirapuri to use the multiple valid field scheme disclosed by Blomgren. Referring to Table 2 again, one condition (00) would indicate invalid state when Blomgren is implemented into Mirapuri since there is no issued instruction in the stage at a certain time and thus the stage is invalid as shown previously. With Blomgren integrated into Mirapuri, the valid bits would be stored

in the register as before and any other bits needed for address tracking may or may not be in the register.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Mirapuri to include the valid field scheme that includes a plurality of valid bits for each instruction as taught by Blomgren so that a low-cost address tracking scheme is realized.

Response to Arguments

25. Applicant's arguments filed 28 October 2004 have been fully considered but they are not persuasive.

26. Applicant has argued with respect to claims 1 and 8 that Mirapuri appears to be silent regarding status information unique to operand data. The included dictionary definition of "unique" (definition 1) shows that to be unique is to be the only one or the sole. Thus the claim language that recites "a first status unique to a first operand data stored in said first register" can be read as a first status of a first operand data stored in said first register, which is the sole status of this type for this operand. With this interpretation one needs only show that Mirapuri discloses that each pipeline stage has only a single status for the instruction (including the operands) therein. As shown in column 9, lines 20-65 Mirapuri only discloses a single chain of registers, one for each stage, indicating the validity of the stage and the instructions and operands therein. Thus, this status which is unique to the operand in what it conveys is disclosed by Mirapuri.

27. Applicant has argued with respect to claims 3 and 10 that Mirapuri is silent regarding transferring operand data from one register to another where the starting operand data is known invalid. The Examiner would like to point out that this is not what claim 3 has been amended to say. Claim 3 instead seems to present an impossible scenario. The claim reads that the first status is set to the invalid state in response to the first register receiving an invalid second operand data prior to transferring the second operand data to the first register. It cannot be understood how a status can be set in response to having received data, yet be set before the data is transferred. Thus the Examiner does not know how to interpret the claim and the argument therewith.

28. Applicant has argued with respect to claim 8 that not motivation appears to exist to modify Mirapuri to stall a processor pipeline in response to a "V" bit as taught by Steiss since Mirapuri already provides a mechanism for stalling a processor pipeline without using a new bit that indicates that data is not available. Indeed Mirapuri does disclose a mechanism for stalling a pipeline as highlighted previously with dependency chain bits. Steiss is simply introduced to show that such a status may be stored in the register with the other data, which is not explicitly given by Mirapuri. The Examiner then showed how one of ordinary skill in the art would have recognized the advantages of a central location for all of this data and thus would have been motivated to move the validity status bits into the pipeline register.

29. Applicant then argues that placing the data and status information in the same register to achieve speedup is merely a conclusory statement and lacks supporting evidence or convincing reasoning. It is completely proper that one of ordinary skill in

the art would expect some sort of speedup from data all being in a central location. This is because the close proximity of the data to the logic that uses it (rather than having the status bits further away from the logic) results in shorter wire lengths for connection. Sending something over a shorter distance is inherently faster than sending something over a longer distance. In addition, one of ordinary skill in the art would have recognized that this close proximity improves area on the chip die since the elements are more compact. Further, the courts have found that shifting the location of parts without modifying the function of a device is not a patentable distinction. See In re Japikse 86 USPQ 70 (CCPA1950).

30. Applicant's arguments, with respect to the rejection of claim 19 have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Blomgren as specified above.

31. Applicant argues regarding claim 6 that the provided motivation for combining is not credited to Mirapuri, BBB, or knowledge generally available to one of ordinary skill in the art and thus is improper. The Examiner is assuming that by BBB, the Applicant means Blomgren, which was used in the rejection. To clarify, one of ordinary skill in the art would have recognized that the ability to only store addresses when needed rather than storing every address at all times is low-cost since fewer resourced are consumed yielding less power consumption.

Allowable Subject Matter

32. Claims 4, 5, 11, 12, 17, 18, and 23 are allowed.

Conclusion

33. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (571) 272-4166. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
January 4, 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100